Stock Return Prediction in the Tokyo Stock Exchange using Machine Learning Methods

Yuhaobai Yuan

{ yuanyuhaobai@gmail.com }

School of Social Science, The University of Manchester, Oxford Rd, Manchester M13 9PL, United Kingdom

Abstract. This paper studies how three machine learning models work in the Japan Exchange Group (JPX) stock market. The models are Ridge regression, Multi-Layer Perceptron (MLP), and LightGBM. It uses past trading data from 2,000 stocks. The study builds a ranking strategy across stocks. The goal is to increase the daily spread return Sharpe ratio. The results show that the linear model (Ridge) and the MLP model do not produce strong risk-adjusted returns. But the LightGBM model performs better. It reaches the highest Sharpe ratio of 0.039272. The study also uses simulation tests. This helps remove forward-looking bias. The Sharpe ratio after this test is 0.028086.

Keywords: Machine learning, Stock market prediction, Japan Exchange Group, LightGBM, Sharpe ratio.

1 Introduction

The ability to predict stock price changes is the base of financial data science, but this task is hard because financial markets have high volatility, they change over time, and the signal in the data is weak compared to the noise. In recent years, many studies have used machine learning methods, and these methods help find complex patterns that traditional economic models often miss. This research studies stock ranking in the Japanese stock market, and it uses the Kaggle Tokyo Stock Exchange prediction dataset, which includes past data for 2,000 listed stocks and gives a strong base to test high-dimensional prediction models.

The importance of this research is that it moves from simple price prediction to portfolio optimization. In a market-neutral strategy, the ranking of assets is more important than predicting exact prices. Traditional measures like Mean Squared Error (MSE) do not focus on risk-adjusted returns, and these returns are very important in real trading. So, this paper uses the intraday spread return Sharpe ratio as the main measure. It focuses on this ratio to find strategies that can create alpha and stay stable in different market conditions, and this helps meet the main needs of long-term algorithmic trading.

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365110

The main goal of this research is to build and test a machine learning pipeline that ranks 2,000 stocks to increase the risk-adjusted Sharpe ratio. To do this, the paper compares three models. The first is ridge regression as a linear model. The second is a Multi-Layer Perceptron (MLP) as a deep learning model. The third is LightGBM as an ensemble model. The method builds daily long-short portfolios, and it buys the top 200 ranked stocks and sells the bottom 200. To make the results reliable, the paper uses simulated streaming inference loops, and these loops avoid forward-looking bias and data leakage, so the model performance reflects real trading limits.

2 Literature review

The evolution of financial forecasting reflects a steady transition from classical statistical frameworks to advance nonlinear computational architectures.

2.1 Transition from traditional to deep learning architectures

Traditional financial models often have difficulty dealing with complex nonlinear relationships in large datasets, so the field is now moving to machine learning (ML) and deep learning (DL) methods. Recent studies show that combining technical analysis indicators with deep learning models like LSTM and GRU can improve the model's ability to follow market trends [1]. The latest review shows that neural networks (NNs) are the most widely used model in stock prediction, and they make up 46.3% of all machine learning research cases. This is mainly because they can handle nonlinear trends and adjust to complex patterns between inputs and outputs [2]. Also, reviews in this area show that nonlinear models are important for solving the limits of traditional statistical methods [3].

2.2 Model fusion and hybrid architectures

To make models more stable, researchers have studied "decision fusion" strategies, and these strategies combine signals from several weak models to build a stronger trading signal [4]. At the same time, advanced frameworks use data cleaning and mixed structures, for example combining CNNs to extract local features and RNNs to learn time patterns, and these methods have improved prediction accuracy [5,6]. In studies of highly volatile stocks, deep learning models, especially LSTM, showed better results than traditional methods like K-NN or linear regression. They reached very high prediction accuracy under different error measures such as SMAPE and RMSE [7]. When the task is set as predicting returns or deciding price direction, the choice of algorithm is still very important for getting stable alpha [8].

2.3 Ensemble methods and performance optimization

Comparative studies show that ensemble methods, such as random forests and boosting, have strong performance in reducing the bias-variance tradeoff in financial time series [9]. These methods work well with nonlinear features and noisy stock market data. But studies also show that model performance can depend on the market index. In some markets, ensemble methods perform better. In other cases, supervised learning models like Logistic Regression can give better results for certain indices such as the NIKKEI 225 or TSX [10]. Also, when the data size becomes larger, deep learning models can find feature levels through multi-layer structures, and this reduces manual work and helps the model stay stable when using long-term historical sequence data [7].

3 Exploratory data analysis (EDA) and data preprocessing

The reliability of financial forecasting models depends on the quality of the data. This study uses a strict data preprocessing process to deal with noise and irregular patterns in stock market time-series data.

3.1 Data cleaning and structural alignment

The first step aligns the time-series data of 2,000 stocks to make sure the time order is consistent.

Handling missing values. Financial data often has gaps because trading may stop or reports may come late. The paper uses a grouped forward-fill method, and it fills missing values with the most recent available value for each stock code.

Target integrity. The paper removes any records without a defined “Target” (forward return), so the loss functions for the Ridge, MLP, and LightGBM models are calculated only with real and confirmed data.

3.2 Feature engineering and technical indicator synthesis

Based on technical analysis ideas, the study changes raw price data into prediction signals. To follow the “no look-ahead” rule, all indicators are built using rolling windows that only use past data.

Moving averages ($\boldsymbol{MA_n}$). These act as low-pass filters, and they reduce short-term noise to show price trends. The n-days moving average for security i at time t is defined as:

$$MA_{i,t,n} = \frac{1}{n}\sum_{j=0}^{n-1} P_{i,t-j} \tag{1}$$

Where $P_{i,t}$ represents the closing price. In this research, it specifically calculates MA_5 and MA_{20} to capture short and medium-term momentum.

Historical volatility ($\boldsymbol{\sigma_{i,t}}$). This metric quantifies risk exposure and price dispersion. It is calculated based on the rolling standard deviation of daily returns over a 20-day window:

$$\sigma_{i,t} = \sqrt{\frac{1}{k-1}\sum_{j=0}^{k-1}\left(R_{i,t-j} - \bar{R}_{i,t}\right)^2} \tag{2}$$

Where k = 20, and $R_{i,t}$ is the daily return.

Daily momentum ($\boldsymbol{R_{i,t}}$). The percentage change in daily closing prices provides a direct measure of the immediate momentum effect:

$$R_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}} \tag{3}$$

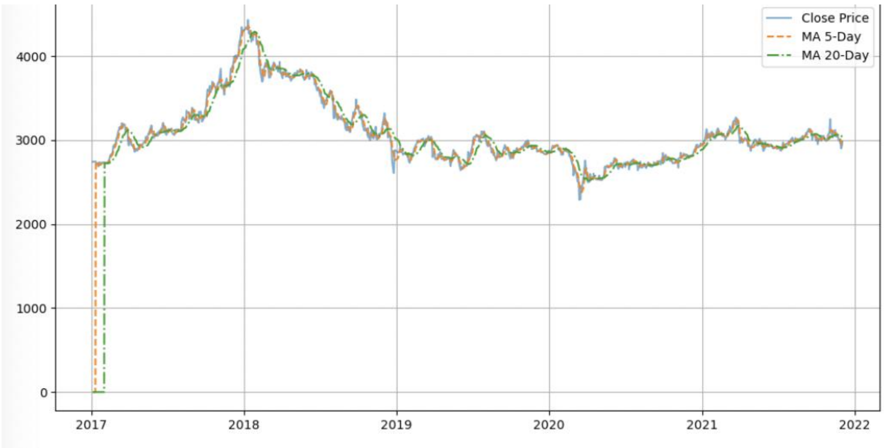

Fig. 1. Stock Price and Moving Averages (Sample: 1301) (Picture credit: Original)

Fig. 1 illustrates the closing prices alongside the 5-day and 20-day moving averages (MA_5 and MA_{20}), calculated as per Eq. (1). Fig. 1 demonstrates that moving averages effectively smooth out market noise to reveal underlying price trends. It can be observed from Fig. 1 that MA_{20} identifies long-term trends while MA_5 captures short-term momentum.

3.3 Statistical normalization and scaling

Machine learning architectures, particularly the Multilayer Perceptron (MLP) and Ridge Regression, are sensitive to the magnitude of input variables.

Z-score standardization. The paper applies a linear transformation to each feature to achieve a mean (μ) of 0 and a standard deviation (σ) of 1:

$$z = \frac{x - \mu}{\sigma} \tag{4}$$

This process ensures that no single feature dominates the model's weight updates due to its original scale.

Table 1. Descriptive Statistics Table

	MA5	MA20	Volatility	Daily Return
count	2332267.00	2332267.00	2332267.00	2332267.00
mean	0.00	-0.00	-0.00	0.00
std	1.00	1.00	1.00	1.00
min	-0.72	-0.72	-0.30	-12.63
25%	-0.44	-0.44	-0.15	-0.16
50%	-0.22	-0.22	-0.07	-0.01
75%	0.12	0.13	0.04	0.13
max	29.06	27.65	131.19	281.63

4 Methodology

This section gives a clear and detailed explanation of the research pipeline, and it includes the math form of the prediction models, the ranking method, and the strategy evaluation framework.

4.1 Predictive model architectures

Based on the statistical results found earlier, this research tests three different types of models to solve the stock ranking problem. These models range from a simple linear regularization model to a deep learning model and an ensemble gradient boosting model.

Ridge regression (linear baseline). Fig. 2 shows the correlation heatmap of the engineered features, and the chart shows strong multi-collinearity among the moving average indicators because there are clusters with high correlation.

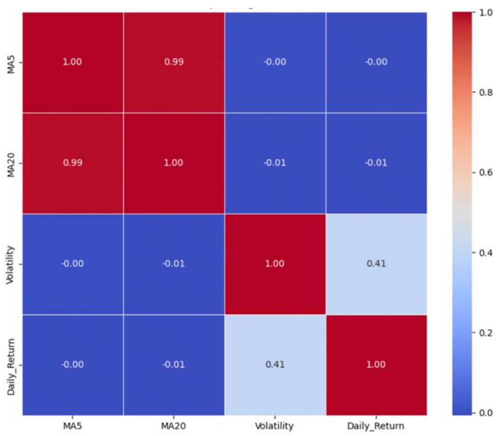

Fig. 2. Correlation Heatmap of Engineered Features. (Picture credit: Original)

To set a performance benchmark and deal with the high multi-collinearity ($r > 0.85$) between MA_5 and MA_{20}, this research uses Ridge Regression. This model adds L_2

regularization by adding to the Ordinary Least Squares (OLS) loss function, and it includes a penalty term to reduce the size of the coefficients:

$$J(\theta) = \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \tag{5}$$

where λ is the regularization parameter that controls the balance between fitting the training data well and keeping the weights small.

Multilayer perceptron (MLP). To study non-linear relationships in the data, the paper uses an MLP model with fully connected layers. Each neuron first does a linear calculation and then applies a non-linear activation function.

$$a^{(l+1)} = \sigma\left(W^{(l)} a^{(l)} + b^{(l)}\right) \tag{6}$$

The paper utilizes the Rectified Linear Unit (ReLU) activation function, $f(x) = \max(0, x)$, to capture complex alpha signals that linear models might overlook.

LightGBM (ensemble learning). The main model in this study is LightGBM, which is a Gradient Boosting Decision Tree (GBDT) framework. It improves the loss function step by step, and it adds weak models each time to predict the errors from the previous models.

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x) \tag{7}$$

Its leaf-wise growth method and strong structure help it handle fat-tailed return distributions that are common in financial time-series data.

4.2 Cross-sectional ranking and portfolio construction

The core of the strategy is to transition from point-wise return prediction to a cross-sectional ranking framework.

Ranking logic. Each trading day t, the models give a predicted score $\hat{y}_{i,t}$ for each of the 2000 stocks, and these scores are then sorted from high to low and changed into ranks $R_{i,t} \in [0,1999]$.

Long-short strategy. A market-neutral portfolio is built by buying the top 200 ranked stocks and selling the bottom 200 stocks.

4.3 Evaluation metric and validation integrity

The main performance measure is the Daily Spread Return Sharpe Ratio, and it shows the risk-adjusted performance of the long-short strategy.

$$Sharpe = \frac{\overline{R_{spread}}}{\sigma_{R_{spread}}} \tag{8}$$

Where $\overline{R_{spread}}$ is the average daily return difference between the long and short portfolios. To keep the results reliable and avoid look-ahead bias, the paper uses a simulated streaming inference loop, and this method makes the model process data step by step, so it follows real trading conditions where future information is not available at the time of prediction.

5 Results and analysis

This section evaluates the real performance of the method mentioned before. The main measure of success is the Daily Spread Return Sharpe Ratio, and it shows the risk-adjusted performance of the long-short ranking strategy.

5.1 Predictive accuracy and model comparison

Fig. 3 illustrates the predictive performance comparison between Ridge regression, MLP, and LightGBM. And it indicates a clear hierarchy in the predictive capabilities of the three architectures.

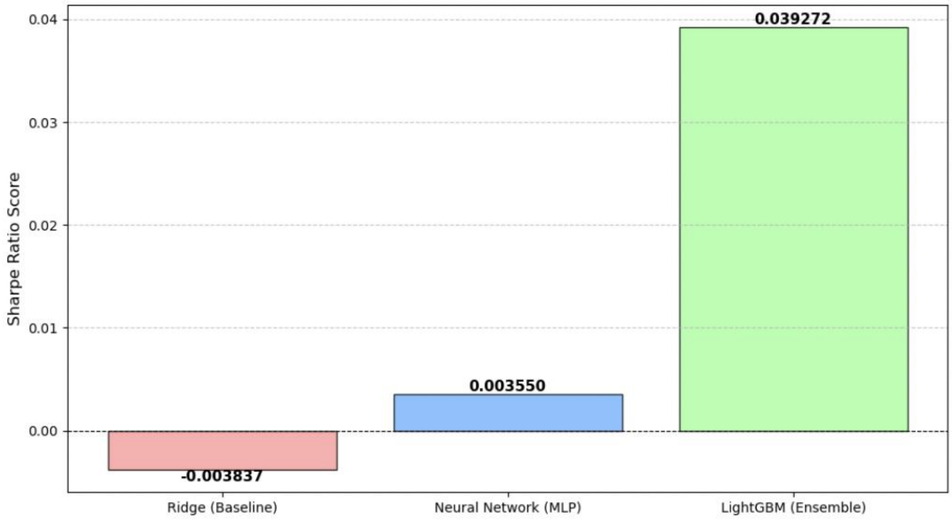

Fig. 3. Model Comparison: Daily Spread Return Sharpe Ratio (Picture credit: Original)

Ridge Regression gave a negative Sharpe Ratio (-0.0038), and this shows that linear models cannot capture the changing and noisy signals of the Tokyo Stock Exchange.

The MLP model achieved a small positive gain (0.0036). Although the non-linear ReLU activation helped improve function fitting, the high noise-to-signal ratio in daily stock returns likely caused overfitting during the training stage.

LightGBM achieved the best performance with a Sharpe Ratio of 0.0392. Its gradient boosting method handled the high-dimensional features well and captured the non-linear relationship between technical indicators and future returns.

5.2 Data distribution and model robustness

The strong performance of LightGBM can also be explained by the statistical features of the processed data. As shown in the descriptive statistics table (Table 1), the target variables show extreme fat tails, and the maximum value of Daily Return (281.63) is much larger than the standard deviation.

Linear models like Ridge are very sensitive to these outliers, and this can distort the regression line. But LightGBM uses a leaf-wise growth method and histogram binning, and these make it more robust to extreme values, so it can keep stable ranking performance across the 2,000 stocks.

5.3 Validation of streaming inference

To make sure the model can work in real trading, the paper used a Simulated Streaming Inference method. After applying this method, the Sharpe Ratio decreased from 0.0392 to 0.0280. Standard batch validation often benefits from accidental data leakage. The streaming loop copies the real rule that only data at time $t-1$ can be used to predict time t. So the LightGBM model still kept a Sharpe Ratio of 0.0280 under these strict streaming rules, and this shows that the strategy can be used in real time without forward-looking bias.

6 Conclusion

This study presented a detailed comparison of machine learning models for stock ranking in the Japanese stock market. It tested Ridge Regression, Multi-Layer Perceptron (MLP), and LightGBM using the JPX Tokyo Stock Exchange dataset. The results show a clear order in prediction performance. LightGBM performed the best and reached a Sharpe Ratio of 0.0392. This is because its gradient boosting method handled the fat-tailed return distributions found in the EDA stage. In contrast, linear models like Ridge Regression could not capture non-linear alpha signals, and MLP models were more likely to overfit when the financial data had high noise.

The research combined technical indicators such as MA_5, MA_{20}, and Historical Volatility, and it showed that trend-following features and risk-level features are important for cross-sectional ranking. The study used a Simulated Streaming Inference method, and it gave a more realistic Sharpe Ratio of 0.0280, and this helped avoid look-ahead bias and time-based data leakage.

Although the ensemble model worked well, this study also admits some limits, and these limits give space for future research.

First, the current model only uses technical indicators from price and volume data. Future research can add other data sources, such as sentiment analysis from financial news or macroeconomic indicators, to better capture the main reasons behind market movements.

Second, although LightGBM worked well, future studies can test more advanced models such as Temporal Fusion Transformers (TFT) or special Recurrent Neural Networks (RNNs), and these models may better capture long-term time patterns in financial time-series data.

Third, the current long-short strategy assumes no transaction costs and perfect liquidity. Future studies should add slippage models and transaction cost analysis, and this will give a more detailed view of real trading profits.

References

[1] Cheng, C. Q., Li, C. Y., & Weng, G. F.: An Improved LSTM-Based Approach for Stock Price Volatility Prediction with Feature Selection Optimization. Artificial Intelligence and Machine Learning Review, 4(1), 1-15. (2023)

[2] Mintarya, L. N., Halim, J. N. M., Angie, C., Achmad, S., & Kurniawan, A.: Machine learning approaches in stock market prediction: A systematic literature review. Procedia Computer Science, 216, 96–102. (2023)

[3] Htun, H. H., Biehl, M., & Petkov, N.: Survey of feature selection and extraction techniques for stock market prediction. Financial Innovation, 9(1), 26. (2023)

[4] Zhang, C., Sjarif, N. N. A., & Ibrahim, R. B.: Decision Fusion for Stock Market Prediction: A Systematic Review. IEEE Access, 10, 81364–81379. (2022)

[5] Kumbure, M. M., Lohrmann, C., Luukka, P., & Porras, J.: Machine learning techniques and data for stock market forecasting: A literature review. Expert Systems with Applications, 197, 116659. (2022)

[6] Vijh, M., Chandola, D., Tikkiwal, V. A., & Kumar, A.: Stock Closing Price Prediction using Machine Learning Techniques. Procedia Computer Science, 167, 599–606. (2020)

[7] Bansal, M., Goyal, A., & Choudhary, A.: Stock Market Prediction with High Accuracy using Machine Learning Techniques. Procedia Computer Science, 215, 247–265. (2022)

[8] Soni, P., Tewari, Y., & Krishnan, D.: Machine Learning Approaches in Stock Price Prediction: A Systematic Review. Journal of Physics: Conference Series, 2161(1), 012065. (2022)

[9] Sonkavde, G., Dharrao, D. S., Bongale, A. M., Deokate, S. T., Doreswamy, D., & Bhat, S. K.: Forecasting Stock Market Prices Using Machine Learning and Deep Learning Models: A Systematic Review, Performance Analysis and Discussion of Implications. International Journal of Financial Studies, 11(3), 94. (2023)

[10] Ayyildiz, N., & Iskenderoglu, O.: How effective is machine learning in stock market predictions? Heliyon, 10(2), e24123. (2024)